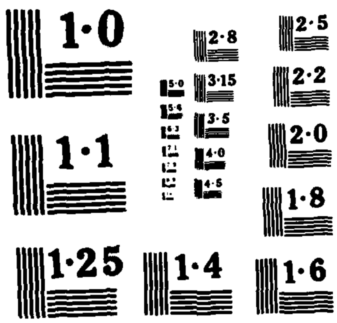
AD-A156 548 COMPUTER-BASED TRAINING FOR THE US COAST GUARD STANDARD 1/1
TERMINAL MICROCOM..(U) NAVAL POSTGRADUATE SCHOOL
MONTEREY CA S E JOHNSON MAR 85

UNCLASSIFIED F/G 9/2 NL


1·0
2·8
2·5
3·15
2·2
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COMPUTER-BASED TRAINING FOR THE U.S. COAST GUARD
STANDARD TERMINAL MICROCOMPUTER:
A BASIS FOR IMPLEMENTATION UTILIZING THE
ELABORATION THEORY OF INSTRUCTIONAL DESIGN

by

Steven Eric Johnson

March 1985

Thesis Advisor: Norman F. Schneidewind

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer-Based Training for the U.S. Coast Guard Standard Terminal Microcomputer: A Basis for Implementation Utilizing the Elaboration Theory of Instructional Design		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis March 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Steven Eric Johnson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE March 1985
		13. NUMBER OF PAGES 80
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A/1	

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer-based training, computer-assisted instruction, instructional design, elaboration theory

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Various factors concerning the U.S. Coast Guard Standard Terminal microcomputer, including the large number of installation sites, steady turnover of user personnel, and conflicting demands of operational commitments indicate that computer-based training is a desirable approach for indoctrinating new users of the computer system. Any such instructional computer program developed for this purpose should consider the characteristics of the trainee audience, particularly the varied (Continued)

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

S N 0102-LF-014-6601

 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT (Continued)

levels of procedural detail desired and constraints on time available for training.
The Elaboration Theory of Instructional Design provides an excellent framework for creating a viable computer-based course of instruction. A Pascal computer program and demonstration lesson modules utilize the macro-strategy components of Elaboration Theory in an introductory course on computer procedures. Course presentation is controlled by special characters imbedded in the lesson text files and interpreted by the program.

Approved for public release; distribution is unlimited.

Computer-Based Training for the U.S. Coast Guard Standard Terminal Microcomputer: A Basis for Implementation Utilizing the Elaboration Theory of Instructional Design

by

Steven Eric Johnson
Lieutenant, United States Coast Guard
B.S., United States Coast Guard Academy, 1976

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March, 1985

Author: Steven E. Johnson

Approved by: Norman F. Schneidewind, Thesis Advisor

Barry A. Frew, Second Reader

Willis R. Greer, Jr., Chairman,
Department of Administrative Sciences

Kneale T. Marshall,
Dean of Information and Policy Sciences

ABSTRACT

Various factors concerning the U.S. Coast Guard Standard Terminal microcomputer, including the large number of installation sites, steady turnover of user personnel, and conflicting demands of operational commitments indicate that computer-based training is a desirable approach for indoctrinating new users of the computer system. Any such instructional computer program developed for this purpose should consider the characteristics of the trainee audience, particularly the varied levels of procedural detail desired and constraints on time available for training.

The Elaboration Theory of Instructional Design provides an excellent framework for creating a viable computer-based course of instruction. A Pascal computer program and demonstration lesson modules utilize the macro-strategy components of Elaboration Theory in an introductory course on computer procedures. Course presentation is controlled by special characters imbedded in the lesson text files and interpreted by the program.

TABLE OF CONTENTS

I. INTRODUCTION

The U. S. Coast Guard makes widespread use of a particular suite of microcomputer equipment, generically referred to as the Coast Guard Standard Terminal. The Standard Terminal is a highly capable machine, with a large variety of system services and applications available through an Executive command interpreter. The Executive offers such diversity and flexibility, however, that a new user of the computer system may be more confused than enlightened by the on-line "help" feature and available documentation. New system users find personal indoctrination in Standard Terminal procedures a necessity.

User training for the Standard Terminal is currently a labor-intensive effort lacking organizational coordination or enforcement of instructional standards. Computer training is made difficult by the large number of Standard Terminal sites, the steady turnover of personnel inherent in a military organization, and conflicting demands on personnel from operational commitments. One possible solution to the Standard Terminal indoctrination problem is the use of computer-based introductory training at all installation sites.

The goal of this thesis is to demonstrate the capabilities of the Standard Terminal in supporting such a

computer-based training application. This is accomplished by meeting several objectives:

- Proposing subject matter content and format for a computer-based training (CBT) introductory course in Standard Terminal operating procedures;

- Constructing a prototype CBT product based on appropriate principles of instructional design and software programming, utilizing the Standard Terminal hardware and software commonly available to field personnel and training support staffs;

- Commenting on the capabilites and limitations of current Standard Terminal configurations in presenting this CBT product.

The scope of this thesis is limited to demonstrating CBT capabilities of the Standard Terminal. No cost/benefit comparison is made between the approach taken here and other instructional media, although observations are presented which justify the consideration of CBT as a viable alternative in training Standard Terminal users.

II. ORGANIZATIONAL CONTEXT

A. BACKGROUND: COAST GUARD STANDARD TERMINAL

In an attempt to standardize many data processing activities and provide field units with access to central databases, in 1981 the U.S. Coast Guard awarded a requirements contract for the purchase of highly capable microcomputers. These Coast Guard Standard Terminals were to be distributed throughout the organization. A key element in the Coast Guard's Information Resources Management Architecture, the Standard Terminal was intended to eventually replace other microcomputer systems in use (through normal equipment attrition and replacement rather than direct substitution), unless a specific user's needs could not be satisfied by the standard computer [Ref. 1]. By early 1984 over 3000 Coast Guard Standard Terminal (CGST) workstations were installed in nearly 1000 sites in the continental United States, Alaska and Hawaii [Ref. 2, 3].

The suite of CGST equipment includes three types of workstation ("AWS," for Applications Workstation; "IWS," for Integrated Workstation; "MWS," for Monitor Workstation), Winchester hard disk and eight-inch floppy disk drives, dot-matrix and daisy wheel ("letter-quality") printers, and

modems.* System equipment configurations may vary considerably among sites. The most common configuration has an IWS or MWS serving as master workstation in a cluster of up to eight terminals (either IWS or AWS). Although each workstation in a cluster has its own Central Processing Unit (CPU) and main memory, the master unit polls the others for their input/output (I/O) requests to secondary memory. Printers and other peripheral devices may be connected to any terminal in a cluster, and be available for sharing by the other workstations. All CGST sites have at least 40 megabytes of hard-disk storage available, and each workstation in the field has at least 512 kilobytes of main memory [Ref. 3].

The present CGST's operating system, CTOS^{tm}, was developed by the equipment manufacturer specifically for this line of computers. User interface with the system is through a commmand interpreter program called the Executive; all system programs and utilities are available through Executive commands. Applications software for the CGST (including periodic updates) is distributed to each site as part of the contracted computer installation order. Included in the "distribution software" are applications for data

---

*The present Standard Terminal workstations and peripheral memory devices are manufactured by Convergent Technologies of Santa Clara, California, and marketed to the Coast Guard by C3, Incorporated of Reston, Virginia.

communications (including terminal emulation and electronic mail), word processing, electronic spreadsheet, and programming tools (text editor, compilers, linker, librarian, etc.). In addition, many operating system services, such as those controlling screen display and disk access, are available through subroutine calls in user programs.

B. STANDARD TERMINAL TRAINING

1. Existing Training

Under terms of the procurement contract, C3, Incorporated is required to provide initial user training at each CGST site. This is done once, at the time of system installation. Subsequent user training is left to local site personnel, typically a designated System Manager. The beginning user is coached one-on-one by an experienced user, or may be exposed to a more structured "classroom" approach. In either case, the training process is labor-intensive. Instructors are frequently diverted from pressing operational obligations to perform this necessary new-user indoctrination. Although many Coast Guard units have established CGST training programs, no formal guidelines for training course content have been promulgated. As a consequence, pertinent material may be presented incorrectly or not at all, depending on the experience and preferences of the instructor.

Once initial instruction is complete, the new user must develop familiarity with the system on his own, aided only by other users or technical documentation which many find obscure and confusing. A hard-copy user tutorial was prepared for the Coast Guard by the Federal Office of Personnel Management and Kinton, Inc., of Alexandria, Virginia, but its scope is limited to system sign-on and the word processor and IQL (Interactive Query Language) applications. The CGST contractor also provides hard-copy tutorials for the word processor and Multiplan[tm] applications. Nevertheless, in the estimation of the Training and Evaluation Division, Coast Guard Office of Personnel, "the Coast Guard has been operating in a [training] survival mode in response to the implementation of the Standard Terminal System." [Ref. 1: p. 27]

Three factors further contribute to the difficulty of providing adequate new-user CGST training: the rate of personnel turnover at CGST sites, the number of sites, and the size of most of these Coast Guard units. As a military organization, the Coast Guard transfers personnel regularly. Any field unit may also make internal reassignments as personnel are advanced in grade or given different responsibilities. As a consequence, the field unit may see a steady flow of service members and civilian employees requiring initial or "refresher" training in use of the CGST. When this turnover is considered with the large number

of CGST sites, the problem of indoctrinating all new users with centralized training resources appears overwhelming.

Most of the Coast Guard units using CGST's do not have the luxury of dedicating full-time personnel to computer system maintenance and training. Billet structures were developed with the assumption that such activity would be handled as a collateral duty (i.e., one responsibility among many assigned to an individual). Since most Coast Guard units have a substantial load of operational tasks to perform, the administrative overhead of CGST indoctrination may be frequently postponed or shortchanged. Operational requirements may also make unit commanders reluctant to send personnel to off-site training courses.

2. A Computer-Based Training Approach

Given the magnitude of the Standard Terminal introductory training problem, a computer-based training (CBT) approach appears reasonable. A computer-based training course is one which is conducted entirely or predominantly through the computer user's interaction with an on-line instructional program. Various authors [Ref. 4, 5, 6, 7] recommend CBT for several reasons:

- There can be "unlimited" simultaneous users of a computer course, each selecting their own area of interest, proceeding at their own pace, and choosing their own time to experience training.

- The same computer equipment used for operations can be used for training; simulation of real-world environments in a training program is possible, and little or no additional investment in hardware necessary. Training is hands-on.

- There are no problems of class scheduling, although terminal scheduling may be a slight local concern. Costs of formal instruction centers and of instructor or student travel/per diem are minimized.

- CBT is easy to update, in whole or in portions, especially when coordinated from a central training facility. Training standardization is improved.

- Impact on operations is minimized, since the student (or his supervisor) can choose his own lesson time; he can be interrupted if necessary, and later resume training at the same point.

- Only required or desired topics need be covered. Since direct application to job requirements can be seen, the user has an active interest in learning the material.

The concept of computer-based training also receives support from Coast Guard directives. One of the explicit goals of the Coast Guard C3/IRM Plan [Ref. 8: p. 1-15] is to "provide systems with minimal training requirements, such as self-teaching or computer-aided instruction...[This] will minimize training cost and lost time and increase use of systems." The Standard Terminal Software Plan also identifies a Coast Guard-wide software need for "interactive training packages for...use of major software packages and utilities," and an "authoring package for development of computer aided instruction systems to support a wide variety of training needs (not just limited to training for operating the Standard Terminal)." [Ref. 9: p. 27]

Computer-based training for new users of the Standard Terminal may not be a perfect solution to the indoctrination problem. Terminals being used for training are unavailable for operational matters. Knowledgeable users will still need to be available to answer unanticipated trainee questions. As with any self-paced instruction, there is no guarantee that trainees will complete the course or give it the attention expected. Nevertheless, a CBT approach to Standard Terminal training does offer advantages in pertinent areas, particularly the logistics of meeting widespread training requirements, and control over standards of training course content.

III. TRAINING NEEDS OF STANDARD TERMINAL USERS

A. USER PROFILE AND TRAINING OBJECTIVES

This thesis focuses on the training needs of new users of the Coast Guard Standard Terminal microcomputer. A "new user" can occupy any pay grade or organizational position in the Coast Guard, so an introductory training course must consider the broadest possible spectrum of individual characteristics (e.g., level of education). There are, however, identifiable commonalities among new users of the CGST: all are adults, and all require access to the CGST to perform some aspect of their jobs. These two traits greatly influence the approach taken by a computer-based training course in operating the CGST (in contrast, for example, to computer-assisted instruction courses which teach children to spell).

There are two further assumptions about the trainee audience which are useful in designing a CBT course for Coast Guard personnel. One is that the trainee is subject to conflicting demands on his or her time; running through a computer training course is a necessary exercise which may be frequently interrupted by more pressing operational demands. Another assumption is that the new user has had minimal exposure to computers, and may even be reluctant to use the machine. Admittedly, not all trainees will possess

these characteristics. Many individuals may want to participate in the course just to refresh skills previously acquired and presently unused. Nevertheless, in preparing for a "worst case" trainee, the CBT course will be more likely to contain content and style applicable to all new users.

The aim of an introductory CBT course on using the Standard Terminal should be twofold.* First, the course must demonstrate the common procedures required for any user to access the computer system and perform productive activities. The other broad objective should be to provide the user with a degree of confidence and flexibility in using the machine. While the first objective involves a straightforward sequence of cause/effect explanations and practice, the second requires an emphasis on "why" certain commands do certain things, in addition to the "how" of the procedures. Is this knowledge important for a new user? Perhaps not right away, but as the user gains experience and comes to rely more on the computer for operational support, the ability to think strategically (as opposed to thinking mechanically) about computer procedures will contribute to both effectiveness and efficiency.

---

*These general course objectives should not be considered a substitute for detailed lesson objectives, as discussed in several sources [Ref. 10, 11].

B. TRAINING COURSE CONTENT

Using the Standard Terminal depends, to a large extent, on knowledge of the commands available through the Executive command interpreter. The procedural portion of the training course will thus focus on using these Executive commands, with supplementary coverage of some physical procedures (e.g., powering up the system, loading floppy disks). Since not all Executive commands are of equal immediate importance to the new user (COPY will see more action than SELECTIVE BACKUP), a hierarchy of importance can be approximated and used to structure lesson emphasis and sequence. Appendix A is a suggested introductory course outline, based on system documentation [Ref. 12], existing lecture course outlines [Ref. 13, 14] and the personal experience of the author in conducting CGST training sessions.* Executive commands which are normally reserved for use by the System Manager, such as those for changing a volume name or assigning directory passwords, are not covered in the introductory course.

Material providing conceptual support for system procedures is interwoven with the procedural portions of the lessons. When considered separately, the conceptual portion

---

*Additional explanation about the structure of the outline in Appendix A will follow in Chapter IV.

of the course constitutes those elements of "computer literacy" most applicable to CGST users, including:

- Data storage and manipulation as electronic/ magnetic digital bits;
- Interpretation of bits and bytes as numbers and alphanumeric characters;
- How data is organized in volumes, directories and files;
- How data is accessed and manipulated by programs in the Central Processing Unit;
- How data is protected, in both a security sense (as with passwords) and a data integrity sense (archiving).

C. TRAINING COURSE STYLE

Zemke and Zemke [Ref. 15] point out several aspects of instructing adult learners which apply to a CBT introductory course in using the Standard Terminal. Adults prefer an efficient, no-nonsense approach toward acquiring some specific knowledge or skill. They also like to decide for themselves the best way to accomplish their learning goal. Information is acquired most readily when it does not conflict with past experience and beliefs,* and is presented step by step at a pace which allows mastery before continuing with new material.

---

*If the possibility of such a fundamental conflict does exist, the instruction must be designed very carefully, providing deliberate, incremental introduction to and substantiation of the new ideas, rather than assuming unquestioning user acceptance of quick summaries.

Desirable attributes of a CBT course can be determined from these learner characteristics. Foremost among these is the maximization of user choice and control over course presentation. When the adult learner has a clear idea of what information is needed to perform a particular task, the CBT program should give the user the opportunity to cover only the desired topic and bypass extraneous material. A heavy front-end load of explanatory or background information which does not directly address the perceived problem may be considered more of a nuisance than a help. The same can be said about plodding through every lesson frame when a review or a lesson summary is all that is desired. The adult user should have the option of deciding just how much detail about a particular topic is covered.

This presumption of user foresight as a justification for user control will not always apply. A trainee may be at a complete loss about where to begin, and require substantial guidance from the CBT program. This need should be accomodated by having the instructional program "default" to a detailed presentation. User-control options can still be provided for those who desire them.

We can assume that Standard Terminal users will participate in an introductory computer course because they need to use the machine in their job. In this situation, we can further assume that the users will be self-motivated to acquire knowledge about the computer. Any "failure" to learn

the material will be reflected in poor real-world performance. Consequently, within-course tests, monitoring of progress, and recording of course scores (which may be legitimately used as instructional techniques within the course) should not be necessary as inducements for proper course completion.

IV. INSTRUCTIONAL DESIGN

A. INSTRUCTIONAL DESIGN APPLIED TO COMPUTER-BASED TRAINING

A computer-based training course composed without the organizing structure and discipline of instructional design principles is a hit-or-miss proposition. As explained by Gagne and Briggs [Ref. 11], designing instruction goes beyond merely outlining topics to be covered. For example, a successful course design will consider the type of knowledge being conveyed: verbal information, attitudes, intellectual skills, motor skills, or cognitive strategies. Each of these categories calls for a unique presentation strategy (e.g., frame sequence, exercises performed). According to Gagne and Briggs, the course should also specifically address the nine instructional events which are present in any learning situation.* By following these and other guidelines, course

---

*Instructional events:
1. Gaining attention
2. Informing the learner of the objective
3. Stimulating the recall of prerequisite learnings
4. Presenting the stimulus material
5. Providing learning guidance
6. Eliciting the performance
7. Providing feedback about performance correctness
8. Assessing the performance
9. Enhancing retention and transfer

See Gagne and Briggs, [Ref. 11: p. 165]

material is presented to the learner in a manner which capitalizes on current understanding of the mental processes involved in knowledge acquisition. Effectiveness and efficiency in imparting knowledge are thus maximized.

In their prescriptions for instructional design, Gagne and Briggs integrate concepts from several distinct areas of learning theory (cognitive, affective, and psychomotor domains). In a similar fashion, the Gagne-Briggs model has been combined with aspects of other models to create a more general synthesis of instructional design theory. The primary components of this synthesis are Merrill's Component Display Theory (CDT) and the Reigeluth-Merrill Elaboration Theory. CDT focuses on micro-strategy, or the instructional techniques used to present a single idea. Elaboration Theory *is an extension of CDT principles to the macro-strategy* level: how a collection of related ideas can best be combined in a single course of instruction.

This thesis deals primarily with an implementation of Elaboration Theory in a computer-based training course. The following paragraphs discuss essential components of the theory. Micro-strategy principles and related considerations of lesson presentation style are then briefly highlighted.

B. MACRO-STRATEGY: ELABORATION THEORY

As discussed by Reigeluth and Stein [Ref. 16], the Elaboration Theory emphasizes organizational aspects of instruction, as opposed to course delivery or course management concerns. Four problem areas of macro-strategy are addressed: selection of specific ideas for instructional delivery, sequencing of ideas in the course presentation, synthesizing individual course ideas with overall course structure and objectives, and summarizing subject-matter content. Elaboration Theory proposes a general model of instructional design, incorporating techniques to deal with each of these problem areas.

The general model centers on a sequence of lesson modules which elaborate on ideas presented in other, preceding lessons.

> The Elaboration Theory of instruction starts the instruction with a special kind of overview of the simplest and most fundamental ideas within the subject matter; it adds a certain amount of complexity or detail to one part or aspect of the overview; it reviews the overview and shows the relationships between the most recent ideas and the ideas presented earlier; and it continues this pattern of elaboration followed by summary and synthesis until the desired level of complexity has been reached on all desired parts or aspects of the subject matter. It also allows for informed learner control over the selection and sequencing of content. [Ref. 16: p. 341]

Reigeluth and Stein employ a zoom-lens analogy to illustrate Elaboration Theory [Ref. 16: p. 340]. The learner starts with a broad picture, develops an understanding of

the major parts and their relationships (but without a lot of detail), then picks one aspect of the broad picture to zoom into focus. After studying this more detailed view, the learner can zoom back out to reestablish perspective about how the newly-acquired details fit into the overall structure. It is also possible to zoom further into the picture for even more details, provided that no detailed level is seen without first viewing the "higher level" picture of which it is a part.

An overview module is given a special name: epitome. The material presented in an epitome is not an abstract summary of all the course content in lower-level modules. Instead, the epitome consists of a few fundamental, representative ideas that are presented in a concrete, application-level manner (e.g., with examples and practice). The learner acquires specific knowledge which is inherently pertinent, and which is also the foundation for knowledge acquired in lower-level elaboration modules. For example, in our CBT situation, learning to use the command LOGOUT is relatively simple, but incorporates procedural steps common to all other Executive commands.

An epitome module with all of its related elaboration modules is known collectively as a lesson set. Any module situated between the highest and lowest levels of the course is both an elaborating module in one set and an epitome module in another. This hierarchy represents a

simple-to-complex range of information; ideas derived from elaboration modules add to rather than explain or clarify ideas given in the epitome.

The Elaboration Theory identifies instructional strategy components in addition to the elaborative sequence, which deals with the main structure of the course. Individual lessons may use:

- Learning-prerequisite sequences, or presentations of the critical components of an idea;[1]

- Content summarizers, which include condensed restatements of the lesson's ideas and facts, reference examples, and practice items;

- Content synthesizers, which provide perspective of how a lesson's content fits into the logical structure of the overall course content;

- Analogies, useful in relating novel ideas to examples with which the learner is familiar;

- Cognitive-strategy activators, either embedded (e.g., graphic diagrams, mnemonic asociations) or detached (hints about useful learning techniques to employ);

- Learner control, particularly over the pace of instruction and the selection and sequencing of lesson content.[2]

---

[1]Gagne and Briggs developed the concept of learning prerequisites, or preliminary knowledge which must by acquired before the subject ideas can be comprehended. In their model, learning prerequisites for a particular intellectual skill can be discovered through learning task analysis, and used to create a learning hierarchy as a guide to lesson sequencing. [Ref. 11]

[2]One study of U.S. Navy enlisted personnel showed 30% to 66% improvement in course completion time of a CBT program when students were given control of branching and the option of bypassing extraneous material; satisfactory completion scores were maintained [Ref. 17]. Tennyson notes, however,

Elaboration Theory discusses course content as well as guidelines for course framework. Subject matter can belong to one of three types of content: concepts, procedures, or principles. Reigeluth and Stein [Ref. 16: p. 343] define a concept as "a set of objects, events, or symbols that have certain characteristics in common," and provide "sonnet" as an example concept. A procedure is "a set of actions that are intended to achieve an end...a skill, technique, or method." A principle is "a change relationship; it indicates that relationship between a change in one thing and a change in something else," and is usually a theoretical description of cause and effect.

Epitomizing and elaboration is done for one particular type of subject matter content. This is necesssary because different types of content may call for different techniques of simple-to-complex sequencing. The structure of the course will reflect the type of content judged to be most important for meeting instructional objectives; the course will therefore have either a conceptual, a procedural, or a theoretical organization. Information in this dominant category is called the "organizing content." Ideas in the other two categories and rote facts may be presented in the

---

that for learner control to be effective, the learner must have timely feedback about his or her progress toward the educational goal [Ref. 18].

course as information relevant to the organizing content. This information is referred to as "supporting content."

An introductory course in using the Standard Terminal will have a procedural organization; the primary objective is to provide trainees with knowledge of the actions they must take to effectively use the CGST. The "computer literacy" portions of the course represent conceptual supporting content. No theoretical content is involved. Appendix A, a proposed CBT course outline, distinguishes between lesson elements which reflect the organizing content and those which are of the supporting type.

C. MICRO-STRATEGY AND PRESENTATION STYLE

Merrill's Component Display Theory is a precursor of Elaboration Theory. CDT offers detailed guidelines for designing instruction to present a single idea. It prescribes techniques for presenting subject matter based on its organizing and supporting content, and matched with the level of learner performance desired. [Ref. 19]

The literature contains many suggestions for creating effective computer-based instructional courses; most are, in actuality, micro-strategy components or techniques. For example, a common differentiation among presentation strategies gives the instructional designer a choice of drill-and-practice, tutorials, simulations, or games [Ref. 20, 21]. Each of these strategies is concerned with

what the learner sees in successive video displays within a particular lesson, and how specific lesson elements can be best presented. They do not address the problems of subject matter selection, proper sequencing of ideas, synthesizing component ideas with overall course structure, or summarizing subject-matter content.

Considerations of presentation style are also micro-strategy components. These include such factors as:

- Making video screen displays easy to read, as by double-spacing all text and using a clear-screen function instead of scrolling the screen display;
- Using video screen graphics to support or replace text;
- Including motivational elements to encourage prolonged learner interest;
- Ensuring a high degree of user interaction with the computer, beyond mere "page-turning" of video screen displays;
- Selecting appropriate types of reinforcement or feedback in response to user input;
- Designing effective and non-intimidating user/computer dialogues (computer prompts; rules of communication). [Ref 20, 21, 22]

While it is beyond the scope of this thesis to cover micro-strategies in any greater detail, they are vital considerations in preparing any CPT product. Component Display Theory, in particular, provides a guiding framework which can incorporate the specific recommendations on presentation style made by other authors.

V. IMPLEMENTATION STRATEGY AND TECHNICAL NOTES

A. COURSEWARE LOGIC

1. Implementation of Elaboration Theory

The "How To" CBT course* on using the Standard Terminal is a fairly direct implementation of Elaboration Theory principles. The course has a procedural organization, although the conceptual supporting content (in essence, the "computer literacy" portion) represents a substantial percentage of all ideas presented. Separate on-line lesson text files correspond directly to the lesson modules outlined in Appendix A. These files can be summoned only in the sequence prescribed for the course's epitomes and elaborations. The trainee, however, has complete control over which elaboration "branches" to pursue. In fact, the availability of elaboration modules throughout the course can be considered as an extended menu of lesson topics.

User control is effected through special keys on the terminal keyboard. With single keystrokes, the trainee is able at any time to advance lesson frames, reverse frame sequence, restart the current frame, restart the current

---

*The generic term applied to a complete computer-based instructional program is "courseware;" this refers to instructional design logic as well as computer program logic.

lesson module, return to the epitome, and change from the "Learn" mode to the "Review" mode (see page 35) or back. When appropriate in the lesson (specifically, when in a Summarizer/Synthesizer frame of an epitome, as explained below), the user can branch to an elaboration module. A prompting message lets the user confirm that the selected elaboration is the correct one. The user can also exit the course at any time by pressing the FINISH key; again, a prompting message allows the user to confirm the FINISH request.

There are two notable differences between this course's design and the general model for instruction prescribed by Elaboration Theory. Reigeluth and Stein recommend that, after an epitome has been mastered and the learner progresses to a more detailed level of elaboration, control of elaborating lesson presentations should chain directly through the sequence of lesson modules at that new level. A within-set Summary/Synthesizer module is interposed between succeeding lesson modules to provide the "zoom-out for perspective" feature of their zoom-lens analogy [Ref. 16: p. 367]. In the "How To" course, however, control does not pass directly from one elaborating module to the next in sequence at the same level. Although each lesson module ends with a Summary/Synthesizer group of lesson frames, lesson control (and video display) returns to the epitome lesson at the point from which the elaboration

lesson was called. Subsequent modules at the elaboration level can then be called as desired from the epitome. This change of tactics not only provides the user with more control over the course, but deals more readily with sharp discontinuities in subject matter between "successive" elaboration modules.

The second disparity between the "How To" course and Elaboration Theory guidelines involves the change in type of organizing content for some elaboration modules. In the general model of Reigeluth and Stein, the organizing content selected for the course (conceptual, procedural or theoretical) must apply to all lesson modules within the course. The introductory CBT course under consideration, however, contains a large portion of conceptual underpinnings for the main topic, computer operation procedures. For example, a fairly detailed discussion of computer hardware concepts is necessary to explain the functions of all the equipment confronting the new user. This amount of detail is presented by allowing procedural-content epitomes to be elaborated upon by conceptual-content modules.

2. Format and Features of the "How To" Course

The "How To" program is summoned by entering the Executive command HOW TO. After a series of time-paced introductory video displays, control of the course (as far as advancing/reversing lesson frames, etc.) is given to the

trainee. In a micro-strategy sense, the course is primarily a tutorial. Lesson frames correspond to pages in a book. After ideas are presented, the user is queried to verify that their content has been comprehended. The program can respond to user input ("answers") in a number of ways: advancing to the next frame, skipping frames, showing an error message and soliciting different input, branching to remediation frames, returning to preceding frames for review, or calling different explanatory lesson modules. The other categories of presentation strategy (drill and practice, simulations, and games) can be incorporated throughout the course to provide presentation techniques appropriate for the subject matter.

No attempt was made to simulate various Standard Terminal application programs (such as Word Processing and the Executive command interpreter) for instructional purposes. Instead, the CBT program provides the capability to branch directly to an application program for "real-world" user practice. When the trainee executes either a normal or abnormal FINISH from the application, the instructional program is automatically reentered at the frame following the point of departure. This strategy eliminates the necessity of reproducing application functions in the instructional program (which, in most cases, would require prohibitive programming efforts). It also lets the trainee practice with the full capabilities

and quirks of the application which he or she will need to use later on the job.*

The user can select either a "Learn" or "Review" study mode. In the Learn mode, all lesson frames are presented; in Review, the user sees only Summary/Synthesizer frames. This allows someone who wants to refresh instruction previously received to bypass the time-consuming detailed steps.

Another program feature is the on-line "help" information, available to the trainee at any time during the course. By pressing the HELP key, the user can see in one screen display the titles of the current lesson and source epitome lesson, the current study mode, and a list of the special key commands for course control. A second consecutive HELP request will route presentation control to a lesson sequence explaining the command-key functions and general course structure.

3. Lesson Control through Text Files

One approach to programming on-line instruction is to establish a separate "run" file for each lesson module (i.e., develop unique source code in a computer programming language to present each lesson module in the course; each lesson is compiled separately and integrated in the course

---

*The primary drawback to this approach, loss of direct CBT program interaction and control, is discussed in a later section.

at runtime through a subroutine call). The "How To" course, in contrast, uses only one "run" file for any number of lesson modules. A set of standard courseware functions, such as advance-to-next-frame or clear-video-display, is written into the program. The lesson module is established as an on-line ASCII text file, with certain characters reserved for program control. The program reads the text file one character at a time, and, when it encounters one of the reserved control characters, performs the indicated function.*

With this control structure, courseware logic is highly flexible; no recompilation of source code is required to effect a change in course presentation. The programming features implemented in the "How To" course make it possible to reproduce each of the three components of structured computer programming (sequence, repetition and choice). The courseware itself can thus conform to currently accepted rules of good programming style.

B. SOFTWARE DESIGN: PROGRAM LOGIC

1. Implementation of Lesson Text File Control

A lesson module text file is accessed by the course program as a read-only input byte stream (an unstructured sequence of byte values). Each byte value in the ASCII file

*This approach is similar to the Altair "Coursewriter" codes used by William K. Jackson, as cited in [Ref. 22].

is examined as it is read, but only the reserved program-control byte values initiate program action. Since the Standard Terminal operating system uses an extended ASCII character set (256 possible values), the full range of standard ASCII characters can be made available for video presentation by ensuring that all control characters have a value greater than 7FH. The control characters presently recognized by the "How To" program are listed in Appendix B.

Program-control byte values may be followed by numeric or character-string "parameters" for use by the program. For example, the code to initiate video character attributes, such as reverse video or blinking, must be followed by a video frame identifier, column number, line number, attribute code, and the number of columns to which the attribute is to apply.

2. Use of Long-Lived Memory

The "How To" program consists of several modules called into main memory as virtual-storage segments. There are a variety of program variables which all subroutines must access, but too many to conveniently pass as subroutine parameters. In addition, there are several program-defined stack structures which require memory storage independent of the particular program module which occupies the swap buffer. To accomodate global program variables and large memory structures, contiguous blocks of system Long-Lived

(LL) Memory* are allocated and initialized at program load. A table is maintained in Block 0 which holds the byte offsets (from the LL Memory base address) of all LL Memory blocks. Consequently, the base address of Long-Lived Memory can be passed as a parameter between program subroutines; a quick table look-up will provide the memory address of any of the stack, table, or variable blocks in Long-Lived Memory, making them available for direct subroutine access.

The stack structures in LL Memory are particularly important to program functions. For example, there is a stack of lesson text file names indicating the "path" the user has taken to call the current lesson module. Returning to a module's epitome is primarily a matter of popping the file name stack, retrieving the new top file name, and opening that file as the input byte stream. Another stack contains last-position markers for each of the files on the text file name stack. These markers indicate the byte offset within the corresponding text file from which the last elaboration call was made. When an epitome text file is

---

*In the Standard Terminal, main memory is allocated for program use as either Short-Lived or Long-Lived. SL Memory begins at the highest available address and expands downward; it is used for all program instructions and system-defined data structures. LL Memory starts at the lowest available address and expands upward; it can be allocated by an application for preserving values after the application itself has terminated. [Ref. 12]

reopened, the program "speed-reads" up to the offset value kept on the stack, then resumes the lesson by checking for program-control characters.

3. Reentrant Features of Course

When the course loads an application program such as the Word Processor, the machine code instructions for "How To" are overwritten in memory and control is completely passed to the new application. However, prior to exiting, the CBT program is able to do two things: establish itself as the "exit run file" for the subsequent application, and preserve a copy of the Long-Lived Memory block contents on disk. When the new application terminates, the "How To" run file, as the declared exit run file, is automatically reloaded. As part of its initialization routine, the CBT program checks the disk to see if a current "SaveLLMemory" file exists. If so, the contents of LL Memory are copied from the disk and program control resumes where it left off (i.e., at the appropriate byte offset location in the current lesson text file).

Returning from the Executive is a slightly different matter since its run file totally resets the application memory space, including the default exit run file. This means that the user must specifically enter the Executive command HOW TO to reenter the course program. Again, the program checks for a current "SaveLLMemory" file

on disk, reestablishes Long-Lived Memory if the file exists, and resumes lesson presentation at the point of departure.

4. Multi-user Environments

The "How To" course is able to run simultaneously on all terminals in a cluster. There is no problem with one workstation examining the "SaveLLMemory" file of another by mistake, then "resuming" the wrong lesson. A different file is created for each workstation, with the file name incorporating the unique workstation number assigned by the master workstation as each clustered terminal is brought on-line. The other factor that provides terminal independence is the opening of each lesson module as a read-only byte stream, as opposed to opening it as an exclusive-use file. Any number of applications can copy read-only files into main memory for program use.

In multi-user environments, it may be possible for one trainee to leave a course in an incomplete state, say by entering the Executive as part of a course "simulation," but logging out and leaving the terminal instead of resuming the course and calling the Finish routine. Anyone else who starts the "How To" program at that terminal will then reenter the course at the point where it was left by the previous user.

This can certainly occcur, but not if at sufficient time has elapsed since the first trainee logged out. The program checks the date and time the "SaveLLMemory" file was

last modified, and, if over four hours previous to the current system date and time, starts the lesson as if the old one had not been left incomplete.

5. Programming Approach

The "How To" source program was written in Pascal. There are two reasons why a multi-purpose programming language was chosen for this courseware over a commercially-available CAI "authoring" language. The first is that, at the time this thesis project was initiated, the U.S. Coast Guard had not selected a service-wide standard authoring language [Ref. 23], and one project objective was to implement a computer-based training course with software commonly available to Coast Guard field units and training support staffs. The other reason is noted by Bork [Ref. 24: p. 203] and other writers. Pascal and other multi-purpose languages provide a high degree of flexibility in defining courseware. There is no inherent bias in the lesson attributable to a particular courseware programmer's assumptions and constraints. In addition, as a "system-standard" language, the Standard Terminal version of Pascal is able to directly access operating system services with subroutine calls. For example, using the CTOS functions ReadByte and WriteByte provides noticeably faster response than the Pascal operations READ and WRITE.

The program is sufficiently large that the major subroutine modules must be brought into main memory as

virtual storage overlays. However, there is no noticeable slowdown in program performance while segment-swapping is performed.* A 128 kilobyte swap buffer is quite adequate for the program's use. To promote modularity of design and reduce the amount of swapping, all keyboard input for course control is handled by one particular module. All program entry and exit (as when chaining to another application run file) is conducted by the main program module, which remains resident in memory while other modules swap.

C. STANDARD TERMINAL CAPABILITIES AND LIMITATIONS

Through CTOS service subroutine calls, the CBT program has extensive control over screen display, timing delays, memory allocations, disk access, keyboard input, file management (creation, deletion, etc.), and a variety of other operating system features. The capabilities these services provide for CBT courseware allow considerable versatility in lesson appearance and function. There are, however, some system limitations worth mentioning.

The characteristics of AWS workstation hardware constitute a significant constraint on the Standard Terminal introductory course. The most apparent limitation is in the

---

*One drawback to using virtual storage overlays on the Standard Terminal system is that PasMin, a special procedure for significantly reducing the size of a run-time task image which meets certain requirements, cannot be used. Since the "How To" program is not intended to be installed as a resident utility, this is not a critical restriction.

video display capability. While IWS and MWS workstations can have up to 132 columns and 34 lines of video display, the AWS is restricted to 80 columns and 28 lines. In addition, the AWS cannot accomodate more than 16 "special characters" (including the graphics characters dicussed below) on any one line of video display. AWS video frames cannot be vertically divided or overlapped. To avoid designing completely different courses for IWS and AWS workstations, the CBT course must cater to AWS specifications. This results in the bottom six lines of video display on IWS terminals being unused, and in restrictions on the complexity of screen graphics.

Although hardware and software modifications for bit-mapped video graphics are an available option for the Standard Terminal, by far the most prevalent graphics capability at field installations is based on the "graphics" characters (i.e., single lines, double lines, and thick lines in varying orientations, intersections, corners, etc.) of the standard video font. The "How To" program relies on this standard font for its video displays. Diagrams are consequently limited to static combinations of "graphics" characters.

There is another restriction on the courseware which stems from characteristics of the current version of the operating system (CTOS 8.0). A preferred approach to conducting "simulations" with other applications would be to

create independent windows in the video display, one under control of the CBT program and another under control of the desired run file. The CBT program could then monitor learner keystrokes in interaction with the "simulated" application, and display appropriate instructional text in its own window. Unfortunately, the application programs of primary interest for such an arrangement (text editor, Word Processor, etc.) each completely reset the system Video Control Block and would overwrite any CBT-prescribed values for video frames (windows). There are other limitations prohibiting this approach which involve the placement and subordination of interactive task images in the primary and secondary application partitions of main memory.

D. SOFTWARE DESIGN: PROGRAM LIMITATIONS

Testing of the "How To" program is incomplete. The courseware was developed as a first-round prototype, on one particular [illegible] Standard Terminal configuration. [illegible] on different hardware types needed, [illegible] of potential lesson-control patterns [illegible]. In addition, the lesson modules used in the initial demonstration version of the CBT course were quite small in size. The estimated hard-disk memory requirement for all the lesson text files of a completed course is on the order of two megabytes; under the program design, any individual lesson file could be as long

as nearly one megabyte. Tests of program response times under these conditions of larger lesson text files is needed.

Some aspects of the present courseware could stand improvement. As discussed above, there is a loss of course control and interaction when an application run file is loaded for "simulation" activity. The trainee must make note of actions he is to take once the subject application is brought into memory; there is no way the CBT program can confirm that these actions are in fact taken, or assist the trainee if he encounters trouble while in the "remote" application.

Analysis of user input is not very sophisticated. The course writer must anticipate acceptable trainee responses to queries (including misspellings, synonyms, etc.) for a direct string-to-string comparison. An "otherwise" option is available for alternate program response (for example, an error message with a request to try the input again), but unanticipated legitimate answers may erroneously result in the "otherwise" set of instructions being executed.

This program is not portable to other makes of computer. It is tightly bound to the current version of CTOS, relying extensively on system-peculiar operating system services. This apparent drawback is counterbalanced by considering that the program is intended to operate exclusively on the Coast Guard Standard Terminal.

Specifications for future replacements for the current CGST hardware/software suite will likely require compatability with existing software, greatly reducing problems of migration to later generations of the CGST.

E. LESSON TEXT FILE "SYNTAX"

Since the lesson text files are in such direct control of the course presentation and program functions, a set of guidelines for lesson-command sequence and appearance (in effect, a programming syntax) must be determined.* Several program commands available to the lesson text file correspond directly to CTOS service calls (e.g., Delay for a specified time interval). These commands must also comply with restrictions of allowable parameter values and formats outlined in the *CTOS Operating System Manual* [Ref. 12]. Although many courseware syntax rules can be implied from examining the program control character descriptions in Appendix B and reviewing existing lesson text files, no formal compilation of such guidelines has yet been produced.

One valuable attribute of the current program is the possibility of including extensive comments in the lesson text files. No special delimiters for comment statements are

---

*An example of one such courseware rule is that an End-of-lesson character should not be preceded by an End-of-frame character; this sequence would necessitate two consecutive NEXT PAGE key entries by the user to continue with the next lesson.

required; the program merely skips over these bytes until a valid control character is encountered. This capability allows clear explanations of course and lesson logic to be imbedded where they are most needed.

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

This thesis demonstrates that a computer-based training course introducing operating procedures and concepts to new users of the Coast Guard Standard Terminal microcomputer is technically feasible, using common CGST distribution software for course development, and applying principles of the Elaboration Theory of instructional design. This particular approach to instruction would provide typical Coast Guard trainees with a high degree of control over course schedule, pace, and subject matter coverage. As a result, training could readily adapt to the unpredictable, high-priority demands of operational requirements.

The "How To" CBT program developed for this project can be readily applied to subject areas beyond an introductory computer procedures course. Course control is invested in any number of unique lesson text files using imbedded program-control characters. The program itself is independent of lesson content and instructional micro-strategy. Any subject matter can therefore be presented by using the established program functions.

All Standard Terminal installations in the field meet certain minimum hardware criteria (e.g., 512 kilobytes of main memory in all workstations; at least AWS video display

capability). This courseware was developed to operate on equipment with the standard configuration. Consequently, the primary constraint for program installation and use is the amount of hard disk storage space available for the lesson text files.

Any software application which can be run on the Standard Terminal can be incorporated as part of the CBT course without the protracted efforts required to develop a simulation. This makes the "How To" course highly adaptable to changes in available software. Another consideration is the migration capability of this particular software product. The CBT program will remain viable as long as future generations of the Standard Terminal provide operating system features compatible with the current version.

Course logic and presentation can be revised quickly, since there is no need to recompile source code. This makes rapid course prototyping and testing possible. It is also easy to accomodate local additions or modifications to the course (for example, to include the locally-defined names of printers or discussions of special utilities). In fact, when the Executive is entered as part of the course, it is possible to modify the lesson text file currently "in use." Even if the tinkering course writer inadvertantly ruins the "current" text file (from a standpoint of course syntax or

logic) when using this capability, no permanent damage is done since the actual program source code is never changed.

There are some drawbacks to the CBT program in its present form. The "How To" course, under the current version of the operating system, is not able to run interactive software applications under CBT program control with simultaneous video displays (windows). Thorough testing of the courseware remains to be conducted, especially in an actual training situation with a "finished" product. Another constraint is the impossibility of anticipating all reasonable trainee questions about course content; local system managers or other knowledgeable users would need to be available to assist trainees when the CBT program does not.*

Course writers may also find that the lesson text files are difficult to prepare and interpret. Each of the 35 program-control characters presently in use must be inserted as hexadecimal byte values through the Standard Terminal's text editor. The video display symbols for these characters (particularly superscripted and subscripted numerals; see Appendix B) can be awkward to identify.

---

*The impact of any computer-based training course on Coast Guard field personnel will also depend on their individual outlooks toward computers in general. Kearsley, Hillelsohn and Seidel note that people with positive feelings about computers benefit from computer-based instruction, whereas those with negative feelings do not [Ref. 25: p. 106].

B. POTENTIAL COURSEWARE ENHANCEMENTS

There are several potential enhancements to the "How To" program:

- An on-line course outline within the "How To" course, similar to the present HELP feature, would help users "lost" in the course, and those searching for a particular skill area to review.

- A special file could be created to record each user session for subsequent replay, if desired. This would be analagous to the ".ts" file created by the system's text editor.

- A resident utility accessing the "How To" program could be developed to act as an extended system HELP feature. A special key input (e.g., ACTION-HELP) could trigger an excerpted tutorial on whatever Executive command name was placed in the "Command" bar.

- An authoring system for this CBT product would be a definite benefit. Rather than juggling hexadecimal codes in the text editor to define lesson control, the course writer would specify textual content, video characteristics, and other course features through an "English-language" interface; program-control codes, lesson text, and comments would then be established in the lesson text file automatically. [Ref. 26]

C. RECOMMENDATIONS

This thesis draws no conclusions about the relative merits of computer-based training and other instructional media. The primary intent of the project is to demonstrate the technical feasibility of CBT for indoctrinating new users of the Standard Terminal, and to comment on some organizational advantages to this form of instruction.

Specific recommendations include:

- When making decisions about an appropriate media for training new users of the Coast Guard Standard Terminal, computer-based training should be considered a technically viable alternative.

- The principles of Elaboration Theory and the capabilities of the "How To" course can be applied as comparison criteria in evaluating commercial computer-assisted instruction authoring systems.

- If found to be useful and cost-effective, the "How To" program and lesson modules can be further developed in their present form for use in the field. There are additional considerations for continuing this project:

- Since the courseware would be installed in existing equipment, cost/benefit analysis of the project should treat the expense of all Standard Terminal hardware and system software as a sunk cost [Ref. 27].

- A team approach to course development, involving specialists in subject matter content, instructional design, presentation media, and course programming will likely result in a more efficient and instructionally sound course than is possible for any single individual to create [Ref. 20, 28].

APPENDIX A

COMPUTER-BASED INTRODUCTORY COURSE OUTLINE

A. OUTLINE STRUCTURE

1. Numbering Scheme - Epitomes and Elaborations

Numbered paragraphs in section B of this Appendix indicate lesson modules. Subsidiary "elaboration" modules for each "epitome" module are themselves epitomes for lower-level lesson modules. Each module number is a concatenation of its epitome's number and a within-set sequence number, separated by a period (for example, module 2.4.1 is the first lesson in the sequence elaborating on module 2.4). An individual module's number will thus reveal its "hierarchy path" and elaboration level.

Organizing and supporting content ideas for each module are lettered and listed immediately after the module name. Supporting content within any module will either be rote facts or ideas opposite in type (procedural vs. conceptual) to the module's organizing content.

Some outline sections have not been detailed. These are indicated by "a..?" in place of specific content ideas, with a parenthetical summary of the module's contents. Specific Standard Terminal Executive commands are shown in all-capital letters.

2. <u>Abbreviations/Symbols Used</u>

(P) Procedural organizing content

(C) Conceptual organizing content

(S) Supporting content

(E) Elaboration provided in lower levels

(L) Depends on locally prescribed system procedures

B. COURSE STRUCTURE

0. MAIN (LEVEL 0) EPITOME (P)

a. Introduction (separate "lesson")

b. Power up / Reset computer (E,L)

c. Set video intensity

- Turn up your workstation (not too far)
- Turn down unused workstations
- Screen tilt and swivel

d. Sign on (E)

e. Use HELP to review/select Executive commands (E)

f. Enter Executive commands; LOGOUT

- Minimum letters to uniquely identify command
- Press RETURN for parameter list or "GO" prompt
- Press CANCEL to cancel command (don't execute)
- Press GO to execute command

g. Reset video intensity / Power down (L)

- Reverse sequence of power up
- Never turn off master workstation when any cluster workstation is in use

1. Power Up / Reset Computer (P)

    a. System hardware (S,E)

    b. Turn on power strip/reset voltage protector (L)

    c. Turn on disk drive, wait (L)

    d. Turn on master workstation, wait (L)

    e. Workstation hardware testing (S,E)

    f. Bootstrapping the operating system (S,E)

    g. System Initialization batch jobs (S,E)

    h. Turn on cluster workstation (L)

    i. Alternative: RESET button (L)

        - Response to system crash

    j. Turn on peripheral devices (E)

1.1. System Hardware (C)

    a. Hardware components (description) (E)

    b. Hardware arrangement (configuration) (E)

    c. Protecting hardware

        - End-user concerns for equipment upkeep

1.1.1. Equipment Description (C)

    a. Workstation hardware (E)

    b. Secondary memory (E)

    c. Printers (E)

    d. Modems

        - Digital/analog signal conversion

1.1.1.1. Workstation Hardware (C)

a. CRT - video display (E)

b. Keyboard (E)

c. Central Processing Unit (E)

d. Types of workstation (E)

1.1.1.1.1. CRT - Video Display (C)

a. Character display matrix

b. Character font

c. Graphics capability

1.1.1.1.2. Keyboard (C)

a. Key groups

b. Keyboard LED's

c. Shift lock: not for some keys

d. Variability of keyboard interpretation (S)

- soft font
- function keys; templates
- coincidence with print wheels

1.1.1.1.3. Central Processing Unit (C)

a. Computer chips and busses

b. Memory

c. Arithmetic-Logic Unit (ALU)

d. Controller

e. Input/Output

1.1.1.1.4. Types of Workstation (C)

a. MWS

b. IWS

c. AWS (Cluster and Standalone)

1.1.1.2. Secondary Memory (C)

a. Disk controller

b. Hard disk

- Description

c. Floppy disk

- Description

d. Magnetic tape

- Description

1.1.1.3. Printers (C)

a. Dot matrix

b. Daisy wheel ("letter-quality")

c. Serial vs. parallel

1.1.2. Typical Coast Guard Standard Terminal Configurations (C)

a. Cluster vs. standalone

b. Shared peripheral devices

c. Example configurations

d. Distance limitations

1.2. Workstation Hardware Testing (C)

a..? (Internal tests performed)

1.3. Boostrapping the Operating System (C)

a. Operating System as a controlling program

b. CPU resident vs. disk resident

c. Trouble indicators on boot

1.4. System Initialization batch jobs (C)

a. Initial batch processing

b. Install spooler and queue manager

c. Locally defined batch jobs

1.5. Operating Peripheral Devices (P)

a. Floppy disk (E)

- Loading
- Red light on latch
- Release
- AWS mini-floppies

b. Magnetic tape (E,L)

- Power
- Loading

c. Anadex printer (E,L)

- Position paper
- Power
- On-line light

d. Diablo printer (E,L)

- Position paper
- Power

e. Prentice-Hall modem (E,L)

f. Ventel modem (E,L)

1.5.1. Floppy Disk (P)

a. Types of floppy disks (S)

b. Description (S)

c. Handling

d. Storage

1.5.2. Magnetic Tape (P)

a..? (Description, operating procedures)

1.5.3. Anadex 9500 Printer (P)

a..? (Operating procedures)

1.5.4. Diablo 1640 Printer (P)

a..? (Operating procedures)

1.5.5. Prentice-Hall Modem (P

a..? (Operating procedures)

1.5.6. Ventel Modem (P)

a..? (Operating procedures)

2. Sign On (P)

a. Filling in "forms"

- Selecting lines in form
  - RETURN, NEXT
  - Up and down arrows
- Entering values
  - Reverse video bar, cursor
  - Upper/lower case tolerance
- Correcting mistakes in entries
  - BACK SPACE, DELETE

b. Enter user name

c. Purpose of user name (S,E)

d. Enter password; overtyped with "####"

e. Purpose of password (more later in course) (S)

f. Enter day, date and time (E)

g. Executing "forms" - press GO

2.1. User Name (C)

a. Allows access to computer system

b. Established by System Manager

c. User profile

- Default path
- Default command file
- Signon text files
- Signon chain files

2.2. Day/Date/Time (P)

a. Allowable formats, sequence

b. Arbitrary initialization of system clock (S)

3. Use HELP to Review/Select Executive Commands (P)

   a. Executive as command interpreter for operating system (S)

   b. Executive Status Frame (S)

      - Purpose
      - Components

   c. "Command" bar (S)

   d. Press HELP to list commands, twice for details

   e. Press HELP with partial entry in Command bar

   f. Effective command file (S)

   g. Reason for "complex" command structure (S)

   h. Select command - what do you want to do?

      - Control the computer (E)
      - Work with files (E)
      - Work with text (including printing) (E)
      - Work with numbers (spreadsheet) (E)
      - Work with records (file/database management systems) (E)
      - Talk to other computers (E)
      - Use a floppy disk (E)
      - Control Executive commands (E)
      - Program the computer (E)

3.1. Controlling the Computer (P)

a. SET TIME

b. SCREEN SETUP (MWS/IWS workstations only) (L)

c. System manager options (summary) (S)

- Font Designer

- Configuration files

3.2. Working with Files (P)

a. PATH

b. Volume/directory/files hierarchy (S)

c. Default path; user's "status board" (S)

- Volume [Sys] and directory <Sys> (S)

d. File names; "[]" and "<>" identifiers (E)

e. FILES

f. Required vs. optional command parameters (E)

g. Wild card (*) for file lists (E)

h. COPY (E)

i. Accessing files in another directory (E)

- Appending passwords to file names

j. "[Confirm each?]" parameter

k. "[Overwrite OK?]" parameter (E)

l. TYPE

m. RENAME

n. DELETE (E)

3.2.1. Naming Conventions (P)

a. No blanks (spaces) in file names; max length; name reflect file contents/use

b. File prefixes and suffixes; conventions; ">" as prefix delimiter

c. File prefix as a subdirectory (S)

d. SET FILE PREFIX

e. Multiple file prefixes

f. Limit on number•files in directory (S)

g. Device identification with "[]"

3.2.2. Command Parameter Entries (P)

a. "?" parameters (yes or no)

b. Default parameter values

c. Multiple parameter entries separated by spaces

d. Use of single quotes, ® in parameter entries

e. List file (@FileName) for parameter

f. Parameter "error" messages

3.2.3. Wild Card (*) (P)

a. Correspondence of wild cards between parameter entries

3.2.4. File Copying (P)

a. LCOPY

3.2.5. Password Protection (P)

a. SET PROTECTION

b. Password protection levels/hierarchy (S)

c. Read and write protection (S)

3.2.6. File Content Manipulation (C)

a. Data storage on disk and in main memory

b. Reading and writing data; where actually done

3.2.7. File Deletion (C)

a. File header

b. Data recovery

3.3. Working with Text (P)

a. APPEND

b. Text as data; ASCII code (S)

- Special characters; EOL, EOF

c. EDIT

d. Protection from disaster (S)

- "Name.ts" file

- "-Old" file; renaming

e. REPLAY

f. WORD PROCESSOR

g. RECOVER

h. Housecleaning - delete "*-Old *.ts"

i. Printing copies of text files (E)

3.3.1. Printing Files (P)

a. FORMAT

b. Use of FORMAT for purposes other than printing

c. PRINT

d. Spooled printing (S)

- Shared peripherals
- Disk buffer area
- Queue; priorities

e. SPOOLER STATUS

f. Print control from application programs (Word Processor, IQL, etc.) (S)

g. MAKE WHEEL SET

h. "Printable" characters vs. video display characters (S)

- Different interpretations of ASCII control characters
- Printer limitations

3.4. Working with Numbers (P)

a. Electronic spreadsheets (S)

b. MULTIPLAN

c. Statistics programs (S)

3.5. Working with Records (P)

a. IQL

b. ISAM

c. SORT

d. MERGE

e. MAINTAIN FILE

f. Database management programs (S)

- CT DBMS
- ReQuest
- R:Base 4000 (MicroRim)

g. Planning programs (S)

- PLANIT

3.6. Talking to Other Computers (P)

a. Data communications (S)

b. Electronic mail (S)

- CT Mail (E)

c. Communications protocols (S)

d. Terminal emulators (S)

e. Terminal emulation/protocol programs

- Hardware requirements (cabling, etc.) (S)
- Asynchronous Terminal Emulator (ATE) (E)
- Multimode Terminal Emulator (MTE) / X.25 Network Access Protocol (E)
- 2780/3780 Remote Job Entry (RJE) (E)
- 3270 Terminal Emulator (E)

3.6.1. Asynchronous Terminal Emulator

a..? (Concepts and procedures)

3.6.2. MTE / X.25

a..? (Concepts and procedures)

3.6.3. CT Mail

a..? (Concepts and procedures)

3.6.4. 2780/3780 RJE

a..? (Concepts and procedures)

3.6.5. 3270 Terminal Emulator

a..? (Concepts and procedures)

3.7. Using a Floppy Disk (P)

a. Archiving data (S)

b. IVOLUME

c. VOLUME STATUS

d. Data grouping: paragraphs vs. sectors vs. pages (S)

e. FLOPPY COPY

f. SELECTIVE BACKUP

g. RESTORE

3.8. Controlling Executive Commands (P)

a. SUBMIT

b. Keyboard input vs. submit file input (S)

c. RECORD

d. STOP RECORD

e. Editing a submit file

f. NEW COMMAND

g. REMOVE COMMAND

3.9. Programming the Computer (P)

a. RUN FILE

b. System control and monitoring of program execution (S)

c. PLOG

d. DUMP

e. Creating a run file

- Choose a programming language (E)
- Edit the source file
- Compile (E)
- Link (E)
- Debug

f. Language interpreters

- BASIC
- CRUN

g. FORMS EDITOR

h. Controlling batch jobs

i. Controlling memory partitions

j. Down the road: "fourth generation" languages/application generators (S)

3.9.1. Programming Languages (C)

a. BASIC

b. FORTRAN

c. COBOL

d. Pascal

e. Assembly Language

3.9.2. Compiling Source Code (P)

a. BASIC COMPILE

b. FORTRAN

c. COBOL

d. PASCAL

e. ASSEMBLE

3.9.3. Linking Object Modules (P)

a. LINK

b. LIBRARIAN

c. Using a list file (@FileName)

APPENDIX B

PROGRAM CONTROL CHARACTERS IN LESSON MODULE TEXT FILES

(Note: items within <> brackets indicate repeating groups)

CODE: BYTE VALUE (HEX)	SYMBOL (CTFont)	PROGRAM MEANING	FOLLOWED BY; (COMMENTS)
81H	$\vert$	Start of lesson	Character string (title)
82H	$\vert_0$	Start/end of char string	Characters (between marks)
83H	$\vert_1$	End of lesson	At least one "dummy" char
84H	$\vert_2$	Start of lesson frame	Imbed level (must be at least 16 bytes since last 84H)
85H	$\vert_3$	End of lesson frame	
86H	$\vert_4$	Elaboration from this lesson frame	Nr file names, <Vid frame line nr, Char string (text file spec), Char string (title)>
87H	$\vert_5$	Summary/Synthesizer frame	0; (first one in lesson must be followed by an 89H)
88H	$\vert_6$	Return Point	Imbed level
89H	$\vert_7$	Initialize video frames	<Line start, Nr lines, Border descrip. code (*table), Border char code (DECIMAL EQUIV!), Border attr code (*table)>, 8AH; (list in vid frame nr sequence: 0 up to 7; Nr lines > 35 will give rest of screen to that frame, no more will init)
8AH	$\vert_8$	End vid frame init list	
8BH	$\vert_9$	Reset frame	Video frame number

CODE: BYTE VALUE (HEX)	SYMBOL (CTFont)	PROGRAM MEANING	FOLLOWED BY; (COMMENTS)
	(superscript)		
8CH	^{0}	Set vid frame for display	Video frame number
8DH	^{1}	Start of text for display	Characters (text)
8EH	^{2}	End of text for display	
8FH	^{3}	Go to last Return Point	Imbed level
90H	^{4}	Put vid frame attributes	Vid frame nr, Col nr, Line nr, Attr code (*table), Nr col's
91H	^{5}	Put vid frame characters	Vid frame nr, Col nr, Line nr, Char string
92H	^{6}	Insert user name in text	(Character appears as part of text for display)
93H	^{7}	Put lesson title	Vid frame nr, Col nr, Line nr
94H	^{8}	Put epitome title	Vid frame nr, Col nr, Line nr
95H	^{9}	Put study mode	Vid frame nr, Col nr, Line nr
	(subscript)		
96H	$_{0}$	Position frame cursor	Vid frame nr, Col nr, Line nr
97H	$_{1}$	Read kbd input: LSTRING	Vid frame nr, Col nr, Line nr, Max nr chars, Disposition code: 0 - Answer 1 - User name 2 - TempLstring 3 - TempFileName
98H	$_{2}$	Anticipated answer(s)	Imbed level, Type code (0 - Lstring, 1 - Word), Nr answers which follow, <Char string/number>
99H	$_{3}$	Unanticipated answer	Imbed level, Type code (0 - Lstring, 1 - Word); (OTHERWISE situation)

CODE: BYTE VALUE (HEX)	SYMBOL (CTFont)	PROGRAM MEANING	FOLLOWED BY; (COMMENTS)
	(subscript)		
9AH	$_4$	Begin imbedded instr's	Imbed level; ("conditional" lesson text file commands)
9BH	$_5$	End imbedded instructions	Imbed level
9CH	$_6$	End answer set	Imbed level (MUST HAVE ONE FOR EACH SET OF ANSWERS)
9DH	$_7$	Read kbd input: integer >= 0	Vid frame nr, Col nr, Line nr, Max nr digits, Min OK value, Max OK value, Disp. code (0 - answer, 1 - TempWord), Error msg code (0 - none, 1 - error msg follows), Char string (msg)
9EH	$_8$	Advance-read keyboard	Disposition code: 0 - check for control key input, "discard" 1 - check for control key input, put in TempWord
9FH	$_9$	Display "Keyboard" text	Vid frame nr; (vid frame must have at least 22 lines!)
C0H		Mark video "key" area	Unencoded key value (DECIMAL EQUIV!), Kbd vid attr code (*table)
C1H		KeysToVid interpreter	Target unencoded key value (DEC. EQUIV; use 126 (7EH) if no target), Kbd vid attr code (*table)
C2H		Delay	Nr 100ms (0.1 sec) intervals
C3H		Beep	

CODE: BYTE VALUE (HEX)	SYMBOL (CTFont)	PROGRAM MEANING	FOLLOWED BY; (COMMENTS)
C4H		Run a lesson file and return to next frame	Char string (Lesson module text file spec)
C5H		Chain to run file	Char string (file spec), Char string (file password)
C6H		Set exit run file	Char string (file spec), Char string (file password)
C7H		Enter VLPB simple param	Parameter nr, Char string
C8H		Exit to <Sys>Exec.run	
F1H		Cursor control practice	(First init Vid Frame 0 with 20 lines; put lesson text in other frames)
F6H		Insert contents of TempLstring memory block in text	(Character appears as part of text for display)

VIDEO FRAME BORDER DESCRIPTION CODES

Top	00H	0
Bottom	04H	4

VIDEO FRAME ATTRIBUTE CODES

None	00H	0
Half-bright	01H	1
Underline	02H	2
Reverse vid	04H	4
Blinking	08H	8

"KEYBOARD" VIDEO ATTRIBUTE CODES

None	0
Half-bright	1
Reverse vid	2
Blinking	3
Reverse-vid flash 3 times, then blink	4

LIST OF REFERENCES

1. U.S. Coast Guard Commandant Notice 5230 of 24 August 1983, Enclosure (1), Standard Terminal Management Plan.

2. U.S. Coast Guard Office of Command, Control and Communications Report, Managing End User Computing: One Agency's Approach, by LCDR Fred N. Squires, 28 May 1984.

3. U.S. Coast Guard Office of Command, Control and Communications, Data Systems Division, Standard Terminal Contracting Officer Technical Representative (COTR), CWO Fred Courtney, in telephone conversation, 17 October 1984.

4. Seidman, Marsha D., "Fulfilling the Promise of an Information Center - Through CBT," Journal of Computer-Based Instruction, Vol 9, Special Issue, pp 53-55, May 1983.

5. Buck, John, and Wright, Barbara, "How to Answer the Question:'What's a CBT?'", Journal of Computer-Based Instruction, Vol 9, Special Issue, pp 34-36, May 1983.

6. Kearsley, G. P., "Designing and Implementing Distributed Education," Training and Development Journal, Vol 35, Nr 12, pp 72-7?, December 1981.

7. Dean, P. M., "Why CBI? An Examination of the Case for Computer-Based Instruction," Journal of Computer-Based Instruction, Vol 4, Nr 1, pp 1-7, August 1977.

8. U.S. Coast Guard Commandant Instruction M3090.1, Subj: C3/IRM Plan (U), 25 May 1984.

9. U.S. Coast Guard Commandant Notice 5230 of 24 August 1983, Enclosure (2), Standard Terminal Software Plan (U).

10. University of Southern California Report TR-65, Specification of Training Objectives for Computer-Aided Instruction, by N.A. Bond, Jr., and J.W. Rigney, June 1970.

11. Gagne, R. M., and Briggs, L. J., Principles of Instructional Design, 2nd edition, New York: Holt, Rinehart and Winston, 1979.

12. Convergent Technologies, Inc., Technical Documentation, Convergent Technologies, Inc., 1981-1984.

13. U.S. Coast Guard Commander, Fourteenth District, User Introduction, unpublished lesson outline, 1984.

14. U.S. Coast Guard Commander, Seventh District, Introduction, unpublished lesson outline, 1984.

15. Zemke, Ron, and Zemke, Susan, "30 Things We Know for Sure About Adult Learning," Training/HRD, Vol 18, Nr 6, pp 46-48, June 1981.

16. Reigeluth, Charles M. and Stein, Faith S., "The Elaboration Theory of Instruction," in Reigeluth, C. M., ed., Instructional Design Theories and Models, Lawrence Erlbaum Associates, 1983.

17. Naval Personnel and Training Research Laboratory Report SRR-73-6, Fixed Sequence and Multiple Branching Strategies in Computer Assisted Instruction, by D.A. Slough, and others, September 1972.

18. Tennyson, R.D., "Interactive Effect of Cognitive Learning Theory with Computer Attributes in the Design of Computer-Assisted Instruction," Journal of Educational Technology Systems, Vol 10, Nr 2, pp 175-186, 1981-1982.

19. Merrill, M. David, "Component Display Theory," in Reigeluth, C. M., ed., Instructional Design Theories and Models, Lawrence Erlbaum Associates, 1983.

20. Crawford, Stewart, A Standard's Guide for the Authoring of Instructional Software, Victoria, BC: Joint Educational Management Research, 1981.

21. Alessi, Stephen M., "Designing Effective Computer Assisted Instruction," American Biology Teacher, Vol 46, Nr 3, pp 146-151,192, March 1984.

22. University of Northern Colorado, Greeley, Report 142, The Microcomputer as an Interactive Instruction System in the Classroom, by David L. Jelden, 1980.

23. U. S. Coast Guard Commander, Thirteenth District, Information Resources Management Staff, CWO Jim Lindeblad, in telephone conversation, 7 November 1984.

24. Bork, Alfred, Learning With Computers, Digital Press, 1981.

25. Kearsley, G.P., Hillelsohn, M.J., and Seidel, R.J., "Microcomputer-Based Training in Business and Industry: Present Status and Future Prospects," Journal of Educational Technology Systems, Vol 10, Nr 2, pp 101-108, 1981-1982.

26. Kearlsey, G. P., "Authoring Systems in Computer Based Education," Communications of the ACM, Vol 25, Nr 7, pp 429-437, July 1982.

27. Pressman, Israel, and Rosenbloom, Bruce, "CAI: Its Cost and Its Role," Journal of Educational Technology Systems, Vol 12, Nr 3, pp 183-208, 1984.

28. Federal Interagency Group for Computer-Based Training, Computer-Based Training Starter Kit, U. S. Department of Commerce National Technical Information Service, May 1983.

BIBLIOGRAPHY

Assessment Systems Corporation Report ONR-ASC-82-01, Development of a Microcomputer-Based Adaptive Testing System; Phase I, Specification of Requirements and Preliminary Design, by David C. Vale, and others, June 1982.

Bell, G.M., and Hagamen, W.D., "A Tutorial System (ATS) for Personal Computers," Journal of Educational Technology Systems, Vol 12, Nr 1, 1983-1984.

Caldwell, R. M., "Guidelines for Developing Basic Skills Instructional Materials for Use With Microcomputer Technology," Educational Technology, October 1980.

Caldwell, R. M., Instructional Design Considerations and Teaching Strategy Implementation Using Microcomputers, Bellingham, WA: Association for the Development of Computer-Based Instructional Systems, Western Washington University, 1980.

Canning, Richard G., "Computer-Based Training for End Users," EDP Analyzer, Vol 21, Nr 10, October 1983.

Cherry, George W., Pascal Programming Structures: an Introduction to Systematic Programming, Reston, VA: Reston Publishing Co., Inc., 1980.

Cohen, V. B., "Criteria for the Evaluation of Microcomputer Courseware," Educational Technology, Vol 23, Nr 1, 1983.

Florida State University Report TM-9, An Overview of Computer-Assisted Instruction for Adult Educators, by Walter Dick, 1 October 1969.

Gagne, R. M., Wager, W., and Rojas, A., "Planning and Authoring Computer Assisted Lessons," Educational Technology, Vol 21, Nr 9, 1981.

Gaynor, P., "The Effect of Feedback Delay on Retention of Computer-Based mathematical Material," Journal of Computer-Based Instruction, Vol 8, Nr 2, November 1981.

Greene, John, and Squire, Enid, Computer Training, From Secretarial to Technical Personnel, Bellingham, WA: Association for the Development of Computer-Based Instructional Systems, Western Washington University, 1982.

Hausmann, L. J., "Effective Feedback in Computer Assisted Instruction," Association for Educational Data Systems, Convention Proceedings, 1981.

Howey, Christopher L., The Effects of Different Feedback Strategies on the Attitude of Students Involved in Computer Assisted Instruction, Wayne State University, 1983.

Jones, P.F., "Four Principles of Man-Computer Dialogue," IEEE Transactions on Professional Communication, Vol PC-21, Nr 4, December 1978.

Kearsley, G. P., "Some Conceptual Issues in Computer-Assisted Instruction," Journal of Computer-Based Instruction, Vol 4, Nr 1, August 1977.

Keller, John M., "Motivational Design of Instruction," in Reigeluth, C. M., ed., Instructional Design Theories and Models, Lawrence Erlbaum Associates, 1983.

Kretz, Donald A., "Measures of ADP Training Effectiveness," Study to Determine Viable Approaches to Computer-User Education at NASA, Education/Research, Incorporated, 1972.

Levin, Henry M., and Woo, Louis, An Evaluation of the Costs of Computer-Assisted Instruction, Eric Documentation Reproduction, May 1980.

Magel, K., "Software Engineering Principles for Courseware Development," Association for Educational Data Systems Journal, Vol 13, Nr 2, 1982.

Maginn, Michael, "Let's Jump Off the Individualized Instruction Bandwagon," Training, Vol 14, Nr 1, January 1977.

Martin, M. A., "Cognitive Styles and Their Implications for Computer-Based Instruction," Journal of Computer-Based Instruction, Vol 9, Special Issue, May 1983.

Naval Personnel and Training Research Laboratory Report SRR-73-19, Post Lesson Remediation and Student Control of Branching in Computer Based Training, by George F. Lahey, and others, April 1973.

Navy Personnel Research and Development Center Report NPRDC-TR-79-23, The Effect of Instructional Presentation Sequence on Student Performance in Computer-Based Instruction, by George F. Lahey, June 1979.

Navy Personnel Research and Development Center Report NPRDC-TR-83-29, Methods for Improving the User-Computer Interface, by P.H. McCann, August 1983.

Parnes, J., "How to Design a Computer Terminal Operations Course," Training/HRD, Vol 19, Nr 5, May 1982.

Pressman, Roger S., Software Engineering - a Practitioner's Approach, McGraw-Hill, 1982.

Reigeluth, C.M., "In Search of a Better Way to Organize Instruction: The Elaboration Theory," Journal of Instructional Development, Vol 2, Nr 3, 1979.

Reigeluth, C. M., "TICCIT to the Future: Advances in Instructional Theory for CAI," Journal of Computer-Based Instruction, Vol 6, Nr 2, November 1979.

Self, J. A., "Student Models in Computer-Assisted Instruction," International Journal of Man-Machine Studies, Vol 6, 1974.

Slack, G. W., "Cost-Effective Goals for CAI Instruction Packages," Study to Determine Viable Approaches to Computer-User Education at NASA, Education/Research, Incorporated, 1972.

Slack, G. W., "Dollar-Effective Methods and Techniques for Serving Computer Users," Study to Determine Viable Approaches to Computer-User Education at NASA, Education/Research, Incorporated, 1972.

Steinberg, E. R., "Review of Student Control in Computer-Assisted Instruction," Journal of Computer-Based Instruction, Vol 3, Nr 3, February 1977.

Su, S. Y. W., and Emam, A. E., "Teaching the Principles of Software Systems on a Minicomputer," Journal of Computer-Based Instruction, Vol 1, Nr 3, February 1975.

Tannenbaum, S. M., "There's More to It Than Programming... and Less," Journal of Computer-Based Instruction, Vol 9, Special Issue, May 1983.

U.S. Coast Guard Commandant Notice 5230 of 25 January 1984, Enclosure (2), Coast Guard Information Center Plan (U).

U.S. Coast Guard Institute Pamphlet No. C3IQL3, Interactive Query Language (IQL) User Guide, September 1983.

U.S. Coast Guard Institute Pamphlet No. CC3UGI, Users Guide, September 1983.

U.S. Coast Guard Institute Pamphlet No. CC3WP4, Word Processor Users Guide, September 1983.

U.S. Coast Guard Office of Command, Control and Communications, Plans and Policy Division, LCDR F. Sutter Fox, in telephone conversation, 31 October 1984.

U.S. Coast Guard Office of Personnel, Planning and Evaluation Division, Personnel Policy Analysis Branch, Dr. Robert Frey, in telephone conversation, 19 November 1984.

University of Southern California Report RR-75-32, Why Things Are So Bad for the Computer-Naive User, by William C. Mann, March 1975.

Wager, Walter, "Design Considerations for Instructional Computing Programs," Journal of Educational Technology Systems, Vol 10, Nr 3, 1981-1982.

Watt, Dan, "The Training Game: Instructing Employees in the Use of Micros Is Becoming Very Big Business," Popular Computing, Vol 3, Nr 4, February 1984.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Superintendent ATTN: Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Department Chairman, Code 54 Department of Administrative Sciences Naval Postgraduate School Monterey, CA 93943	1
4.	Professor Norman F. Schneidewind Code 54Ss Naval Postgraduate School Monterey, CA 93943	1 *
5.	LT Barry A. Frew Code 54Fw Naval Postgraduate School Monterey, CA 93943	1
6.	Commandant (G-TPP/HRM) U. S. Coast Guard Washington, DC 20593	2 *
7.	Commandant (G-PTE-1) U. S. Coast Guard Washington, DC 20593	1
8.	Commandant (G-PTE-4) ATTN: Nancy Allinson U. S. Coast Guard Washington, DC 20593	2 *
9.	LT Steven E. Johnson c/o Commander (dt) Second Coast Guard District 1430 Olive Street St. Louis, MO 63103	3

*Copies of software (including source code) provided

END

FILMED

8-85

DTIC